



Title	Proper Interval Graphのランダム生成と列挙
Author(s)	齋藤, 寿樹
Description	ERATO 세미나2010 : No.5. 2010年7月2日
Relation	2010年度科学技術振興機構ERATO湊離散構造処理系プロジェクト講究録. p.31-37.
Issue Date	2011-06
Doc URL	https://hdl.handle.net/2115/48480
Type	conference presentation
File Information	05_all.pdf



ERATO セミナ 2010 - No. 05

Proper Interval Graph のランダム生成と列挙

齋藤寿樹

JST ERATO 湊離散構造処理系プロジェクト

2010/7/2

概要

Interval Graph は、スケジューリング問題やバイオインフォマティクスなど、いくつもの応用を持つことが知られている。そのため、Interval Graph についての研究はいくつも存在し、また Interval Graph 上のアルゴリズムが開発されている。実際の応用で開発されたアルゴリズムを適用する場合、そのアルゴリズムは理論的に効率がよいだけでなく、実装上効率がよいことを実験的解析によって示す必要がある。実験的に解析をするとき、テストデータに偏りがある場合、正しい解析結果を得ることができないため、偏りのないテストデータを生成する必要がある。本発表では、Interval Graph の部分クラスである Proper Interval Graph をランダム生成や列挙するアルゴリズムを提案する。ランダム生成アルゴリズムは数え上げを用いたアルゴリズムで、 n 頂点の連結な Proper Interval Graph を一様ランダムに生成する。列挙アルゴリズムは逆探索に基づいたアルゴリズムで、 n 頂点の連結な Proper Interval Graph を漏れなく、重複なく出力する。



齋藤研究員

Random Generation and Enumeration of Proper Interval Graphs

Toshiki Saitoh (ERATO)

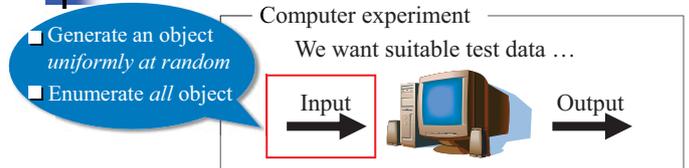
Joint work with

Katsuhisa Yamana, Masashi Kiyomi, and Ryuhei Uehara

第5回ERATOセミナー
2010年7月2日

1

Motivation



Input graphs

- Permutation graphs
- Interval graphs
- Bipartite permutation graphs
- Proper interval graphs (P.I.G)

Random generation
Enumeration

2

Our Algorithms

Random Generation

- Input: Natural number n
- Output: Connected P.I.G. of n vertices
 - Uniformly at random
 - Using a counting algorithm
 - $O(n+m)$ time (m: #edges)

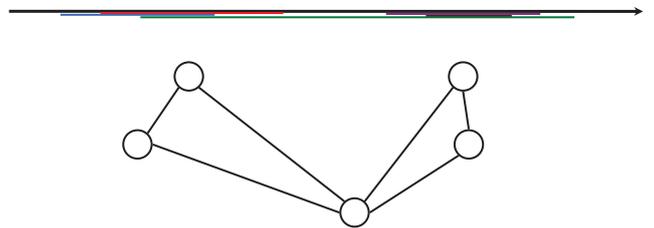
Enumeration

- Input: Natural number n
- Output: All the connected P.I.G. of n vertices
 - Without duplication
 - Based on reverse search algorithm
 - $O(1)$ time/graph

3

Interval Graphs

- Have interval representations

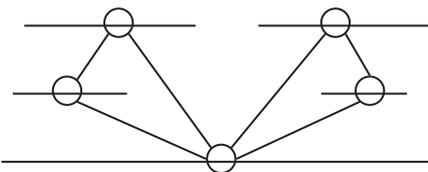


4

Proper Interval Graphs

- Have unit interval representations

Every interval graph has a string representation

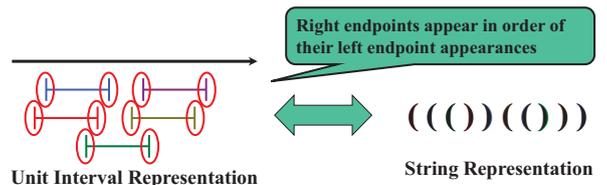


5

Definition

String Representation

- Encodes a unit interval representation by a string
 - Sweep the unit interval representation from left to right
 - Left endpoint \rightarrow "(" : left parenthesis
 - Right endpoint \rightarrow ")" : right parenthesis

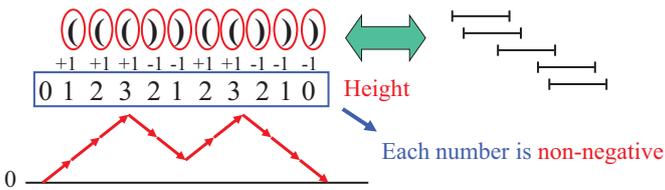


6

Height = # "(" - # ")"

String Representation

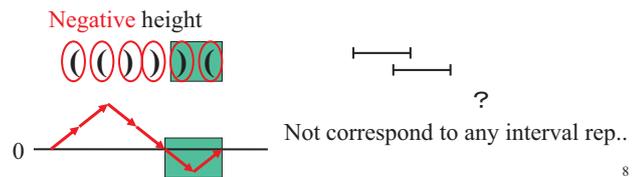
- Property of string rep. of P. I. G. of n vertices
 - Number of parentheses: $2n$
 - Number of "(" : n Number of ")" : n
 - Non-negative
 - Each left parenthesis exists in the left side of its right parenthesis



7

String Representation

- Property of string rep. of P. I. G. of n vertices
 - Number of parentheses: $2n$
 - Number of "(" : n Number of ")" : n
 - Non-negative
 - Each left parenthesis exists in the left side of its right parenthesis

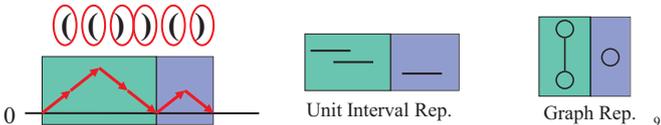


8

String Representation

- Property of string rep. of P. I. G. of n vertices
 - Number of parentheses: $2n$
 - Number of "(" : n Number of ")" : n
 - Non-negative
 - Each left parenthesis exists in the left side of its right parenthesis

Each component corresponds to each area that is bounded by 2 places whose heights are 0.



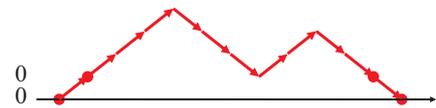
9

String Representation

- Observation 1
 - String rep. of connected P. I. G.
 - Have exactly 2 places whose heights are 0.
 - The left end and the right end

The string excepted both ends parentheses is non-negative

(((())) (()))

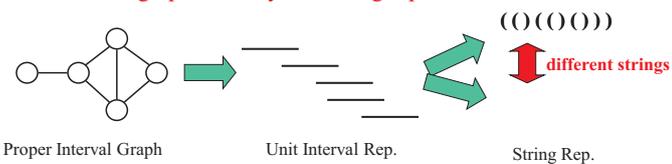


10

String Representation

- Lemma 1. (X. Dell, P. Hell, J. Huang, 1996)
 - A connected P. I. G. has only one or two string rep.

This graph has only two string representations.

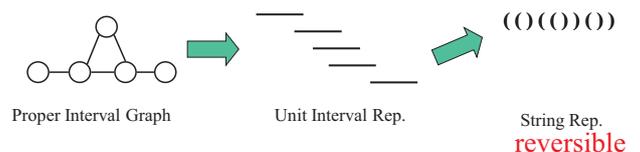


11

String Representation

- Lemma 1. (X. Dell, P. Hell, J. Huang, 1996)
 - A connected P. I. G. has only one or two string rep.

This graph has only one string representation.



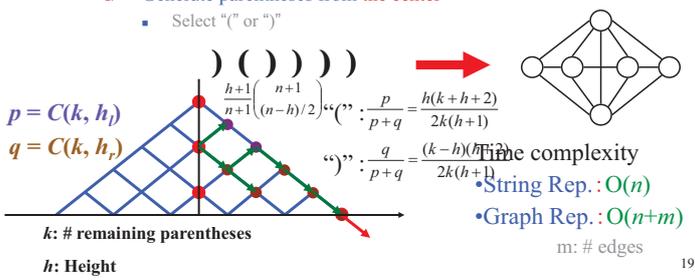
12

Generalized Catalan Number

$$C(n,i) = \frac{i}{2n+i} \binom{2n+i}{n}$$

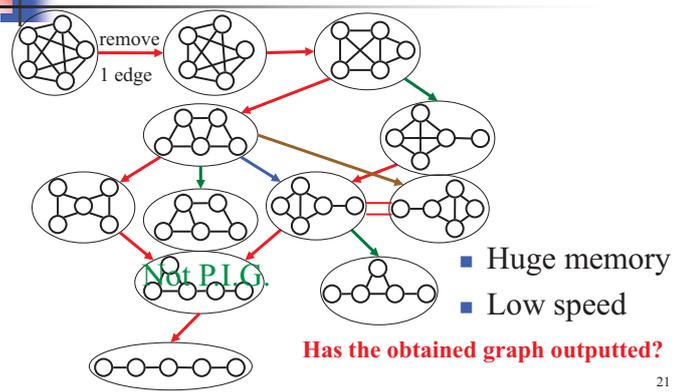
Case 2

- Generation of reversible string uniformly at random
 - Generate a half of the string from the center to the right end
 - Choose the height at the center
 - Generate parentheses from the center
 - Select "(" or ")"

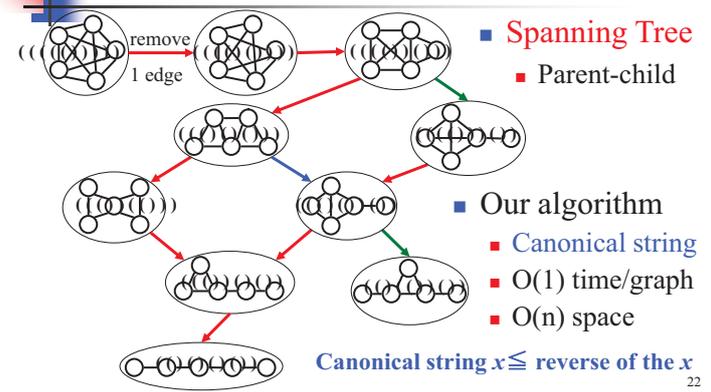


Enumeration Algorithm of Proper Interval Graphs

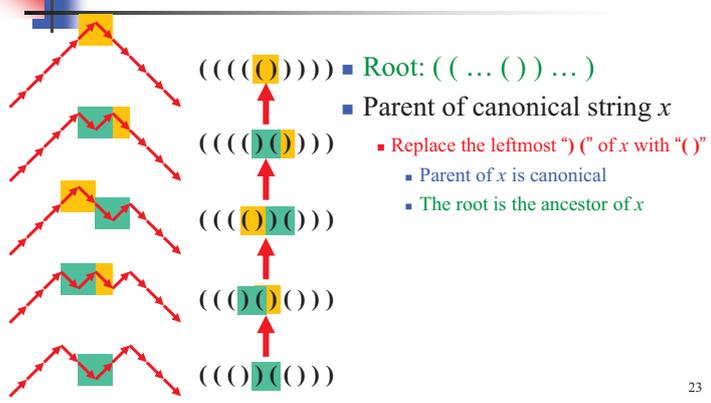
Simple Enumeration Algorithm



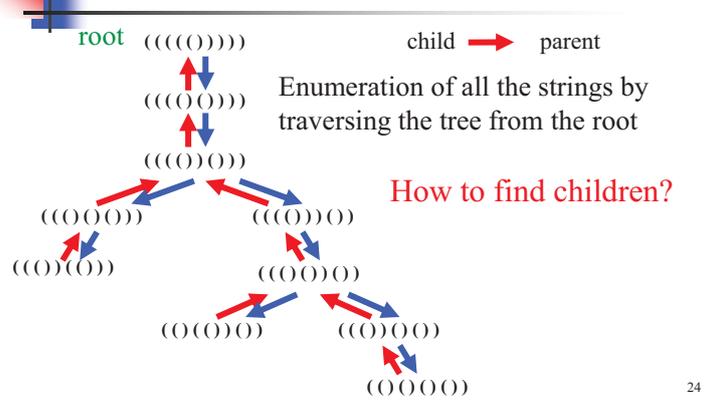
Reverse Search Algorithm



Parent-Child Relation

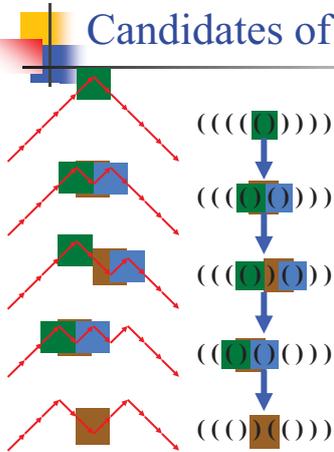


Tree ($n=5$)



- Parent of canonical string x
 - Replace the leftmost "(" of x with "("

Candidates of Children



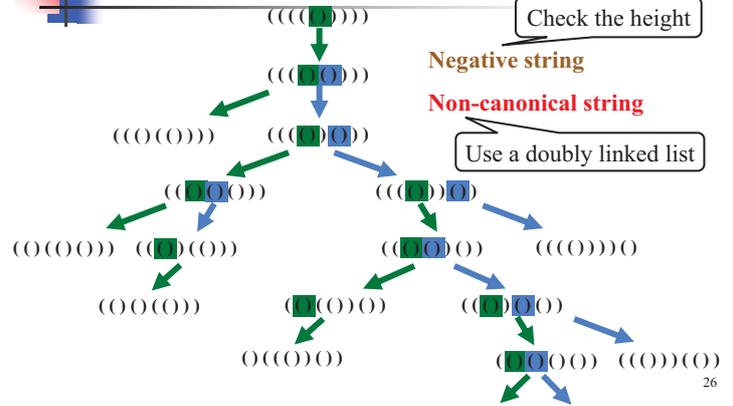
- Child of canonical string y
 - Replace "(" of y with "(" ("
 1. The leftmost "(" , or
 2. If the parenthesis after the leftmost "(" is "(" then "(" can be the candidate
- Next candidates
 - Neighbors of the replaced parentheses

Find candidates: $O(1)$ time

25

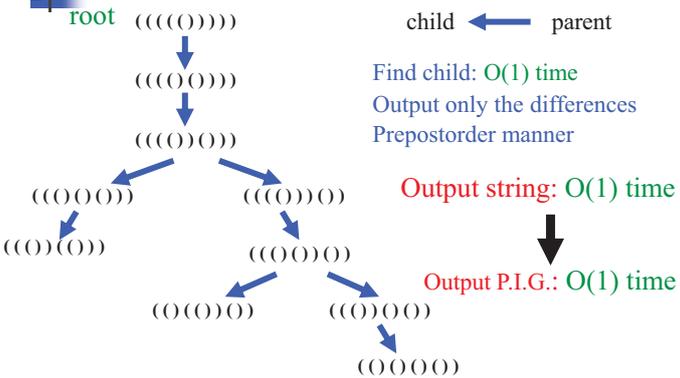
Simply Execution

Check child: $O(1)$ time



26

Tree ($n=5$)



27

Conclusion and Future Work

- Random Generation and Enumeration of Connected Proper Interval Graphs of n vertices
 - Random Generation: $O(n+m)$ time
 - Enumeration: $O(1)$ time/graph
 - n vertices \Rightarrow at most n vertices
- Random Generation and Enumeration of Interval Graphs

28